

Signature Widget

The signature widget is a canvas for freehand drawing, that can be used for capturing signatures. The path data can then be exported.



Creating a Signature Widget

A signature widget can be created by dragging it out from the Tools Palette, where it appears with the following icon:



Alternatively it can be created in script using:

```
create widget as "com.livecode.widget.signature"
```

Using the Signature Widget

Once the signature has been drawn on the widget, the data can be exported using the `pathData` property. This is a numerically keyed array consisting of the individual pieces of path data that make up the drawn path.

Each path info element contains three keys:

- path - SVG path instructions
- color - the color to paint the path with
- width - the line width of the path

The following code can be used to export the `pathData` array to SVG file format:

```

function svgFileData pSignature
    local tPathData
    put the pathData of pSignature into tPathData
    local tSVG
    put "<svg xmlns=" & \
        quote & "http://www.w3.org/2000/svg" & quote & \
        ">" & return into tSVG
    repeat for each element tElement in tPathData
        put "<path d=" & quote & tElement["path"] & quote after tSVG
        put " stroke=" & quote & \
            "rgb(" & item 1 to 3 of tElement["color"] & ")" & quote \
            after tSVG
        put " stroke-opacity=" & quote & \
            round(item 4 of tElement["color"] / 255) & quote after tSVG
        put " stroke-width=" & quote & tElement["width"] & quote \
            after tSVG
        put ">" & return after tSVG
    end repeat
    put "</svg>" after tSVG
    return tSVG
end svgFileData

```

The `pathData` can also be used to manipulate the appearance of the signature. For example, to thicken the line data, the following handler could be used:

```

command thickenSignature pSignature, pFactor
    local tPathData
    put the pathData of pSignature into tPathData
    repeat for each key tKey in tPathData
        multiply tPathData[tKey]["width"] by pFactor
    end repeat
    set the pathData of pSignature to tPathData
end thickenSignature

```

The results of applying this to a signature with factors of 1.5 and 0.5 can be seen below:



Similarly, colours can be applied.

